# Министерство образования и науки Российской Федерации

# Федеральное государственное бюджетное образовательное учреждение

# высшего профессионального образования

# «Ижевский государственный технический университет имени М. Т. Калашникова»

# Кафедра «Программное обеспечение»

Отчет по лабораторным работам

по дисциплине «Конструирование ПО»

тема «Социальная информационная сеть для высших учебных заведений»

Выполнил

студент гр. Б08-191-2 В.М. Воронков

Принял В.Г. Власов

Ижевск 2016

СОДЕРЖАНИЕ

1.ВВЕДЕНИЕ

1.1 Назначение системы

1.2 Область применения системы

1.3 Определения, акронимы, аббревиатуры

1.4 Ссылки

1.5 Обзор системы

2. ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ

2.1 Системный контекст

2.2 Режимы и состояния системы

2.3 Основные функциональные возможности системы

2.4 Основные условия системы

2.5 Основные ограничения системы

2.6 Характеристики пользователя

2.7 Допущения и зависимости

2.8 Оперативные сценарии

3. ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ, УСЛОВИЯ И ОГРАНИЧЕНИЯ СИСТЕМЫ

3.1 Физические

3.1.1 Конструкция

3.1.2 Износостойкость

3.1.3 Адаптируемость

3.1.4 Условия окружающей среды

3.2 Рабочие характеристики системы

3.3 Безопасность системы

3.4 Информационный менеджмент

3.5 Работа системы

3.5.1 Эргономика системы

3.5.2 Ремонтопригодность системы

3.5.3 Надежность системы

3.6 Стратегия и регулирование

3.7 Устойчивость жизненного цикла системы

4. МОДУЛИ ПРОЕКТИРОВАНИЯ

5. UML-ДИАГРАММА КЛАССОВ

6. ТЕСТОВЫЕ ДАННЫЕ

**1. ВВЕДЕНИЕ**

**1.1. Назначение системы**

Разрабатываемая система предназначена для:

* Повышения коммуникабельности между студентами и преподавателями высших учебных заведений;
* Своевременного обеспечения студентов и преподавателей информацией связанной с процессом обучения в высшем учебном заведения;
* Систематизация и структуризация информации связанной с процессом обучения в высшем учебном заведении;
* Автоматизации документооборота между студентами и преподавателями высших учебных заведений.

**1.2. Область применения системы**

Система ориентирована на использование в высших учебных заведениях для обеспечения высокой коммуникабельности, быстрого и удобного информационного потока, автоматизации документооборота между студентами и преподавателями.

Система включает в себя базу данных для систематизации и структуризации хранимых данных и WEB-интерфейс пользователей: студентов и преподавателей.

Основные функциональные возможности системы:

* Коммуникация (приватные переписки, форумы)
* Обеспечение информацией (расписания, новости, оповещения)
* Автоматизация документооборота

**1.3 Определения, акронимы, аббревиатуры**

Пользователь - конечный пользователь системы (студент, староста группы, преподаватель, администратор).

API (англ. Application Programming Interface) - интерфейс прикладного программирования.

ORM (англ. Object-Relational Mapping) - технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования.

**1.4 Ссылки**

**1.5 Обзор системы**

Система делиться на три главных модуля:

* БД

Представленная в объективной форме совокупность данных, систематизированных таким образом, чтобы эти данные можно было быстро найти и обработать;

* API БД

Набор классов, процедур, функций, структур и констант, предоставляемых сервером для взаимодействия с БД;

* Пользовательский WEB-интерфейс

Структура из \*.html, \*.js и \*.css файлов.

**2. ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ**

**2.1 Системный контекст**

**2.2 Режимы и состояния системы**

Система может находиться в следующих состояниях:

Состояние API БД:

* запрос БД и ожидание ответа
* обработка ответа
* вывод на пользовательский интерфейс результатов обработки ответа

Состояние авторизации:

* ожидание авторизации пользователя
* пользователь успешно авторизирован

**2.3 Основные функциональные возможности системы**

Основные функциональные возможности системы

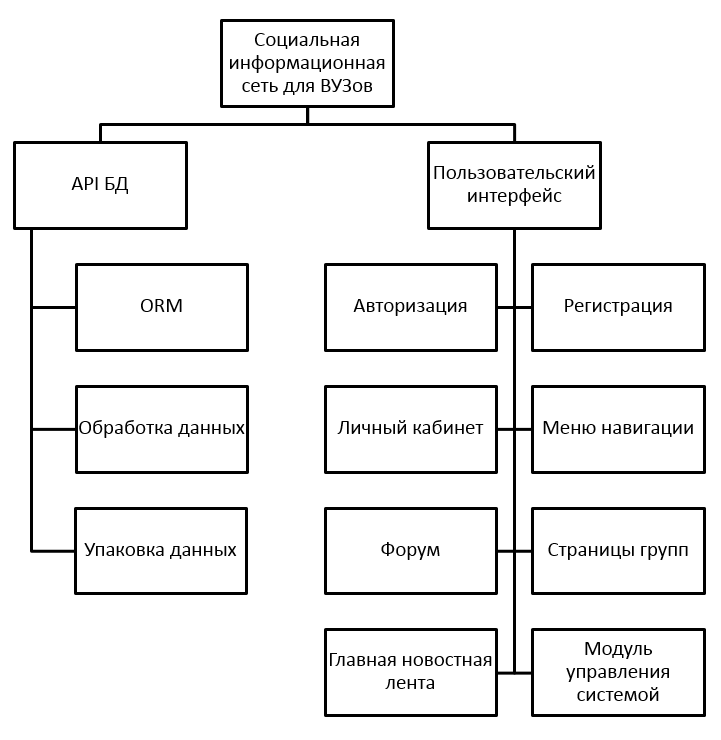


Рис. 1

**2.4 Основные условия системы**

Пользователю необходимо иметь подключение к сети интернет для использования WEB-интерфейса.

**2.5 Основные ограничения системы**

**2.6 Характеристики пользователя**

* Студент

Имеет минимальные права доступа, ограниченные группой. Может использовать приватную переписку, форумы, скачивать и загружать файлы, добавлять новости в группу.

* Староста группы

Имеет права студента, расширенные: полным доступом к группе (редактирование/удаление данных).

* Преподаватель

Права доступа распространяются на кафедру и преподаваемые дисциплины. Может редактировать свое расписание, ставить пометки на документы, использовать приватную переписку, форумы, скачивать и загружать файлы, добавлять новости в группы и на кафедру

* Модератор

Имеет дополнительные права доступа.

* Администратор

Имеет полные права доступа.

**2.7. Допущения и зависимости**

**2.8. Оперативные сценарии**

В системе предусмотрена группа сценариев для различных пользователей – студентов, старост групп и преподавателей.

Все сценарии включают в себя успешную авторизацию, после которой пользователь попадает в личный кабинет. В зависимости от имеющихся у пользователя прав доступа, последний может работать с некоторыми публичными данными. Все пользователи могут просматривать все публичные данные. Приватные данные доступны только владельцам (например, доступом к приватной переписке обладают только переписывающиеся), за исключением администратора.

Администратор может менять права доступа пользователей.

**3. ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ, УСЛОВИЯ И ОГРАНИЧЕНИЯ СИСТЕМЫ**

**3.1. Физические**

**3.1.1. Конструкция**

**3.1.2. Износостойкость**

**3.1.3. Адаптируемость**

В случае увеличения нагрузки на сервер, количество арендованных серверов может быть увеличено, а нагрузка равномерно распределена между ними.

**3.1.4. Условия окружающей среды**

**3.2. Рабочие характеристики системы**

**3.3. Защита системы**

Система должна быть защищена от sql - инъекций и xss - атак.

**3.4. Информационный менеджмент**

**3.5. Работа системы**

**3.5.1. Эргономика системы**

**3.5.2. Эксплуатационная технологичность**

**3.5.3. Надежность системы**

Система защищена от sql - инъекций и xss - атак. Закрытая регистрация (требуется подтверждение модератора/администратора). Доступ к БД только через API.

**3.6. Стратегия и регулирование**

**3.7. Устойчивость жизненного цикла системы**

**4. МОДУЛИ ПРОЕКТИРОВАНИЯ**

Схема модулей проектирования

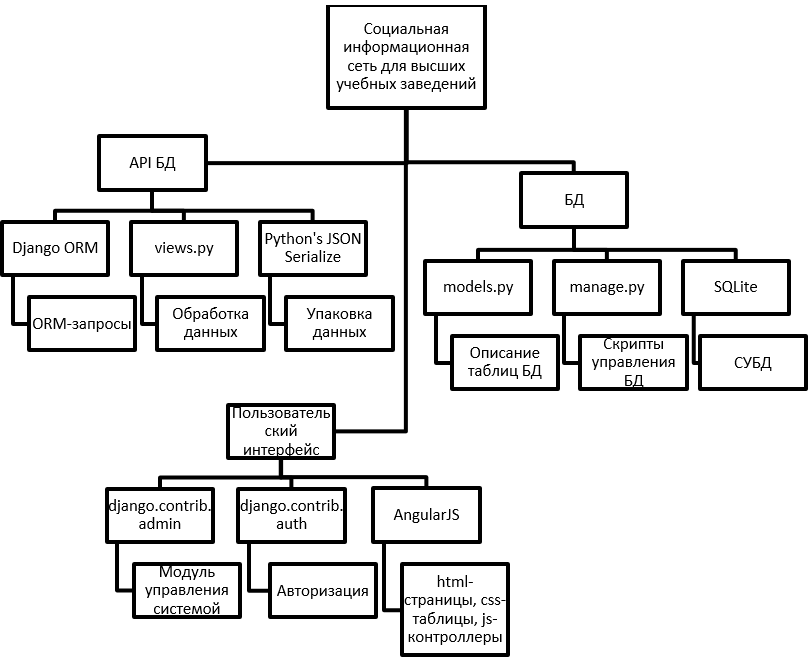


Рис. 2

Классы models.py:

1. Раздел форума

class ForumSection(models.Model):

title = models.CharField(max\_length=200)

2. Подразделение

class Subunit(models.Model):

name = models.CharField(max\_length=60)

upper\_subunit = models.ForeignKey("self", related\_name='lower', null=True)

forum = models.ForeignKey(ForumSection, unique=True, null=True)

3. Профиль пользователя

class UserProfile(models.Model):

user = models.ForeignKey(User, unique=True, related\_name='profile')

subunit = models.ForeignKey(Subunit, null=True)

birthday = models.DateField(null=True)

description = models.TextField(null=True)

4. Комментарий

class Comment(models.Model):

author = models.ForeignKey(User)

massage = models.TextField(null=True)

created = models.DateTimeField(auto\_now=True)

upper\_comment = models.ForeignKey('self', related\_name='lower', null=True)

5. Сообщение в блоге

class BlogMassage(models.Model):

massage = models.TextField(null=True)

created = models.DateTimeField(auto\_now=True)

author = models.ForeignKey(User)

root\_comment = models.ForeignKey(Comment, unique=True)

6. Новость

class News(models.Model):

massage = models.TextField(null=True)

created = models.DateTimeField(auto\_now=True)

root\_comment = models.ForeignKey(Comment, unique=True)

author = models.ForeignKey(User)

subunit = models.ForeignKey(Subunit)

7. Тема на форуме

class ForumTheme(models.Model):

section = models.ForeignKey(ForumSection)

title = models.CharField(max\_length=200)

author = models.ForeignKey(User)

root\_comment = models.ForeignKey(Comment, unique=True)

created = models.DateTimeField(auto\_now=True)

active = models.BooleanField(default=True)

fixed = models.BooleanField(default=False)

8. Предмет

class Subject(models.Model):

name = models.CharField(max\_length=200)

description = models.TextField(null=True)

lecturer = models.ForeignKey(User)

subunits = models.ManyToManyField(Subunit)

9. Занятие

class Lesson(models.Model):

lecturer = models.ForeignKey(User)

subject = models.ForeignKey(Subject)

datetime = models.DateTimeField(null=True)

auditorium = models.CharField(max\_length=10)

template = models.BooleanField(default=False)

10. Мероприятие

class Event(models.Model):

datetime = models.DateTimeField(null=True)

title = models.CharField(max\_length=200)

description = models.TextField(null=True)

subunit = models.ForeignKey(Subunit)

**5. UML-ДИАГРАММА КЛАССОВ**

UML-диаграмма классов

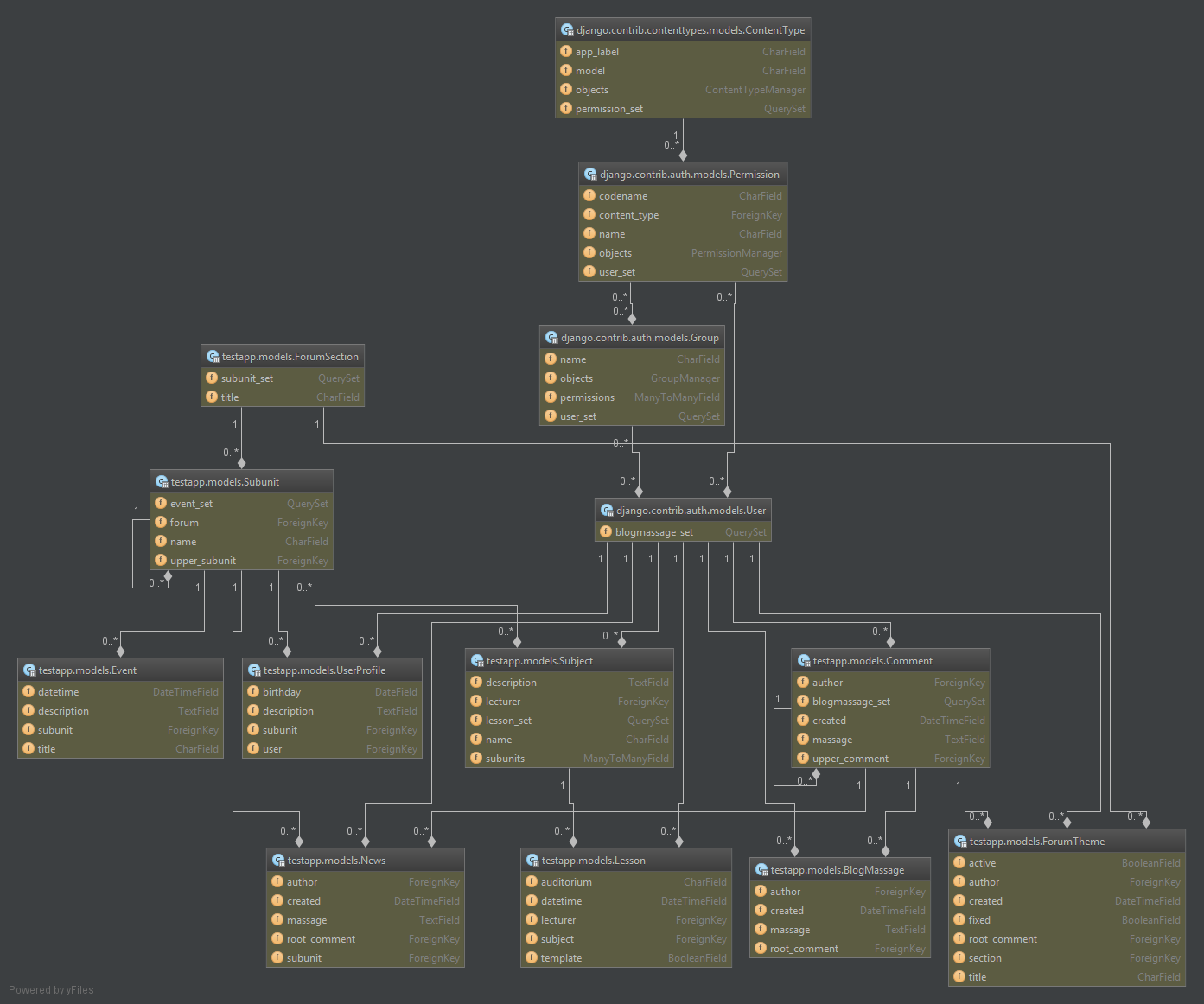
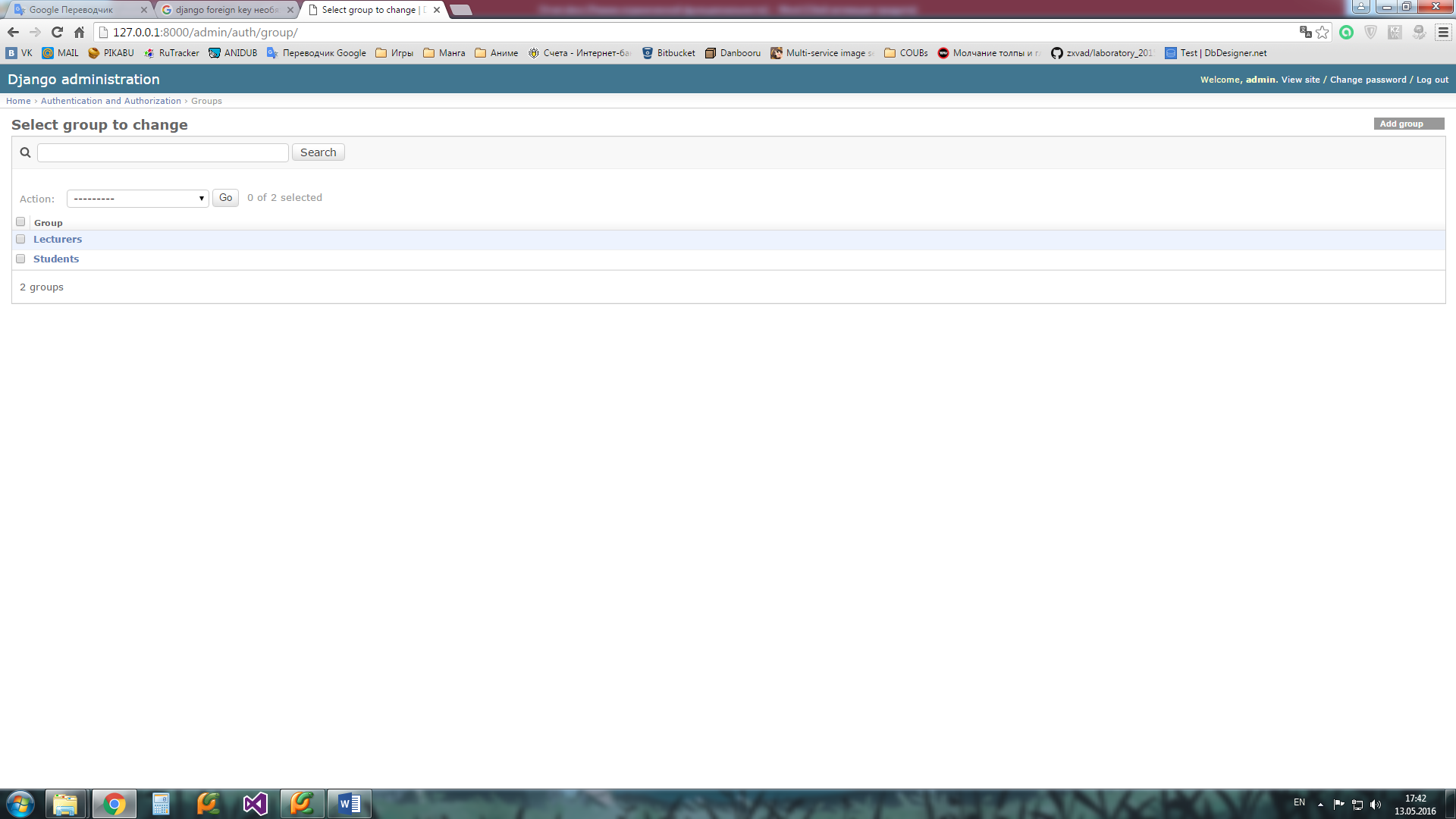
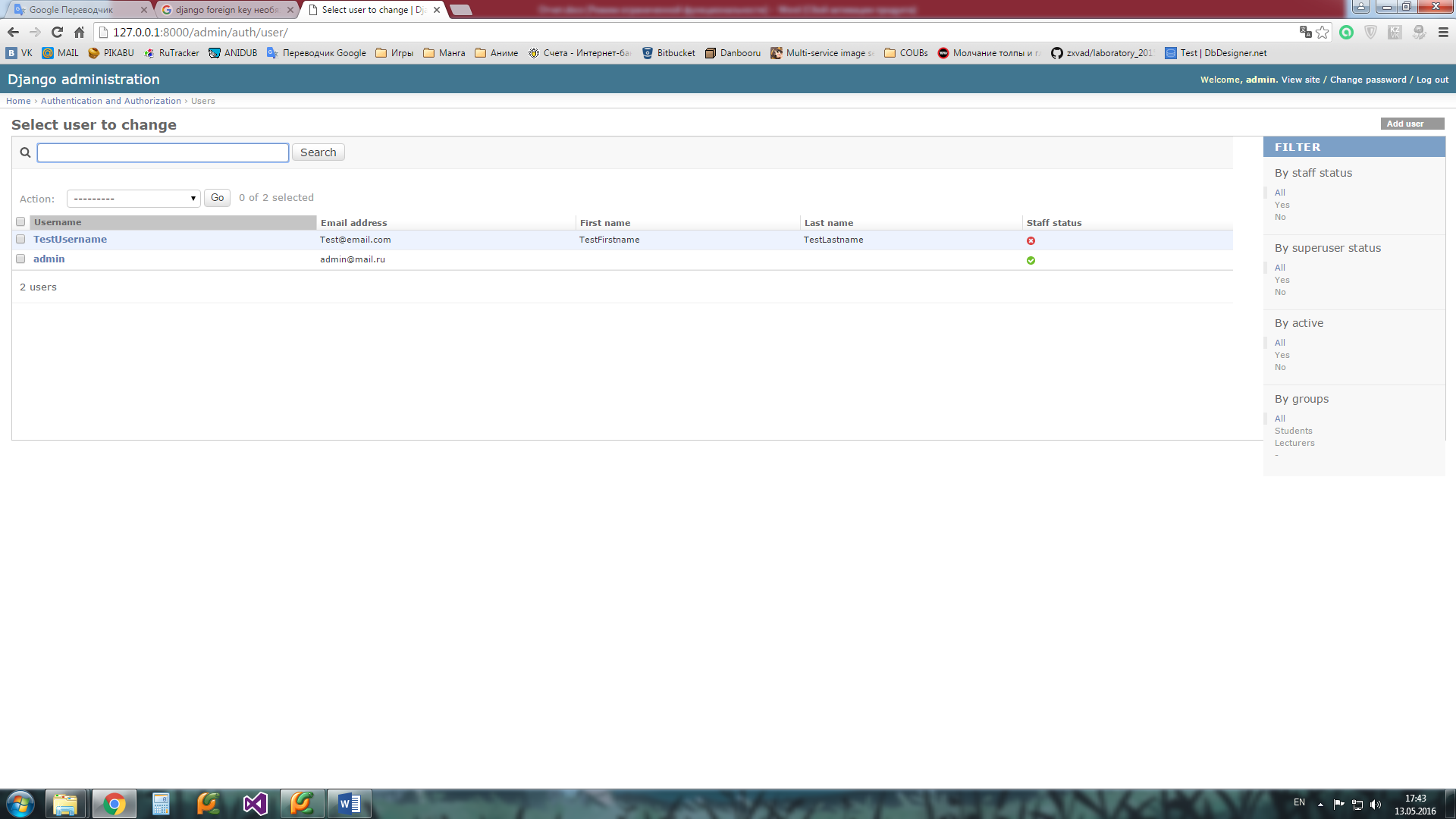


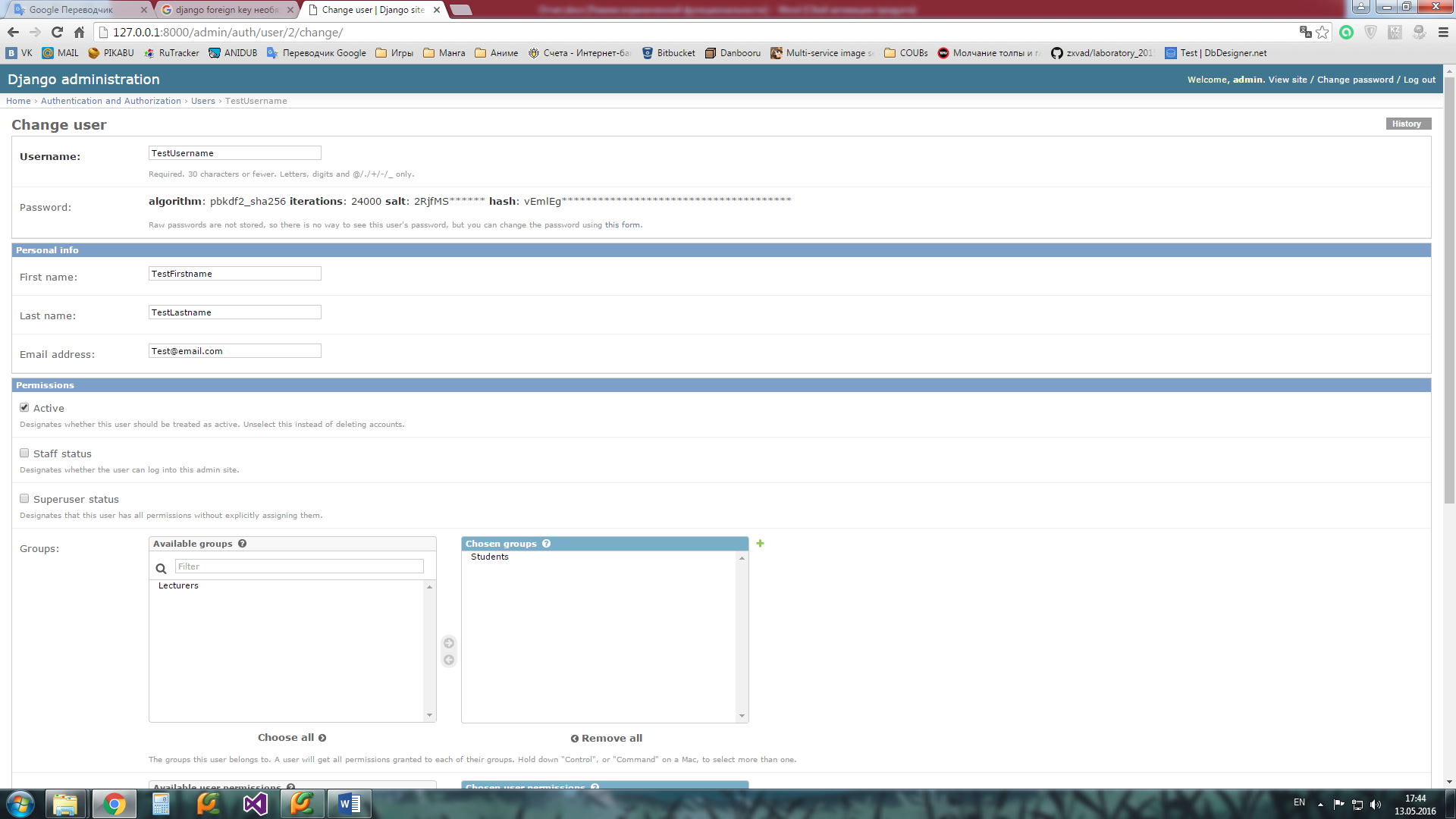
Рис. 3

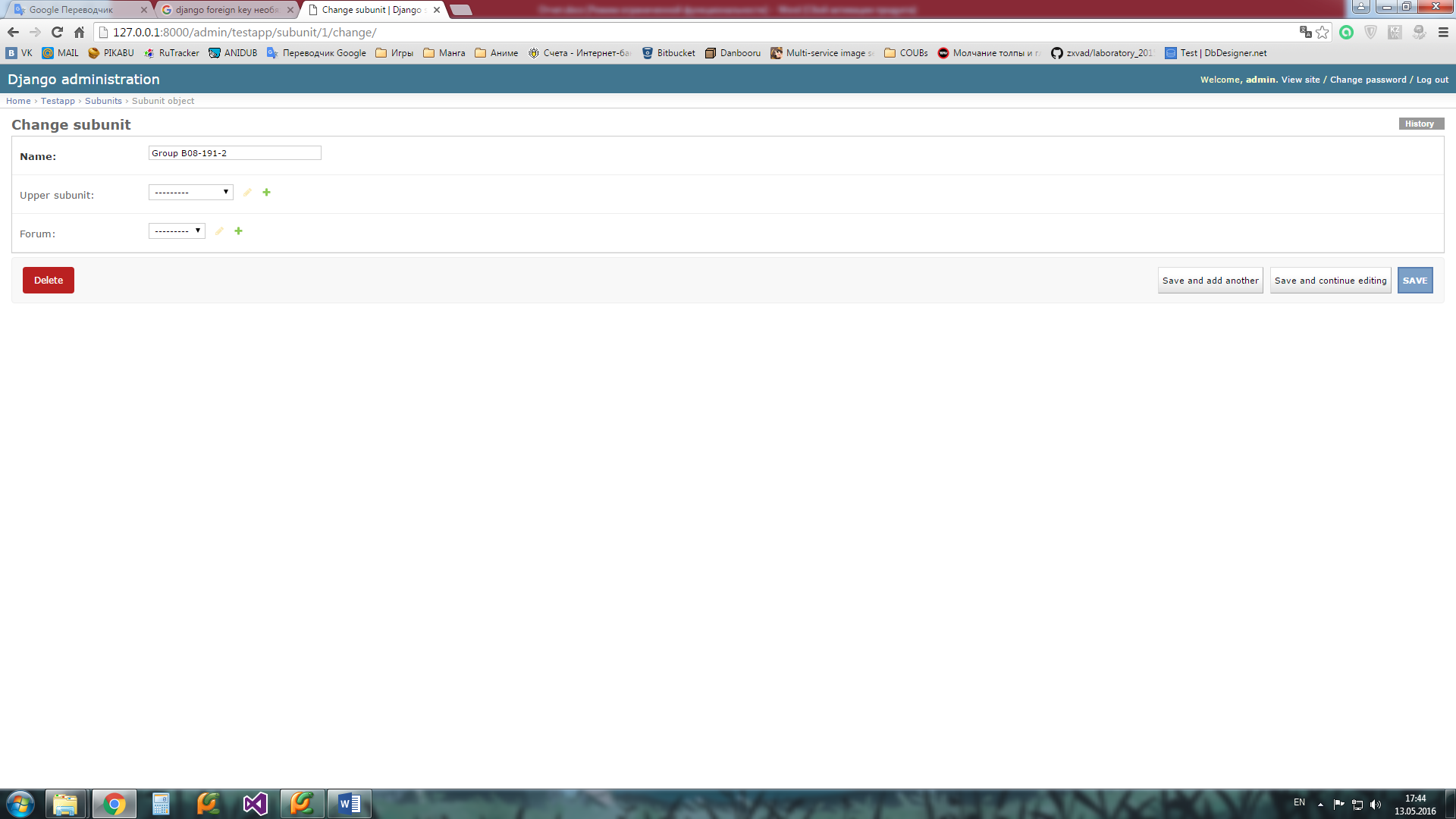
**6. ТЕСТОВЫЕ ДАННЫЕ**

Добавление в БД тестовых данных









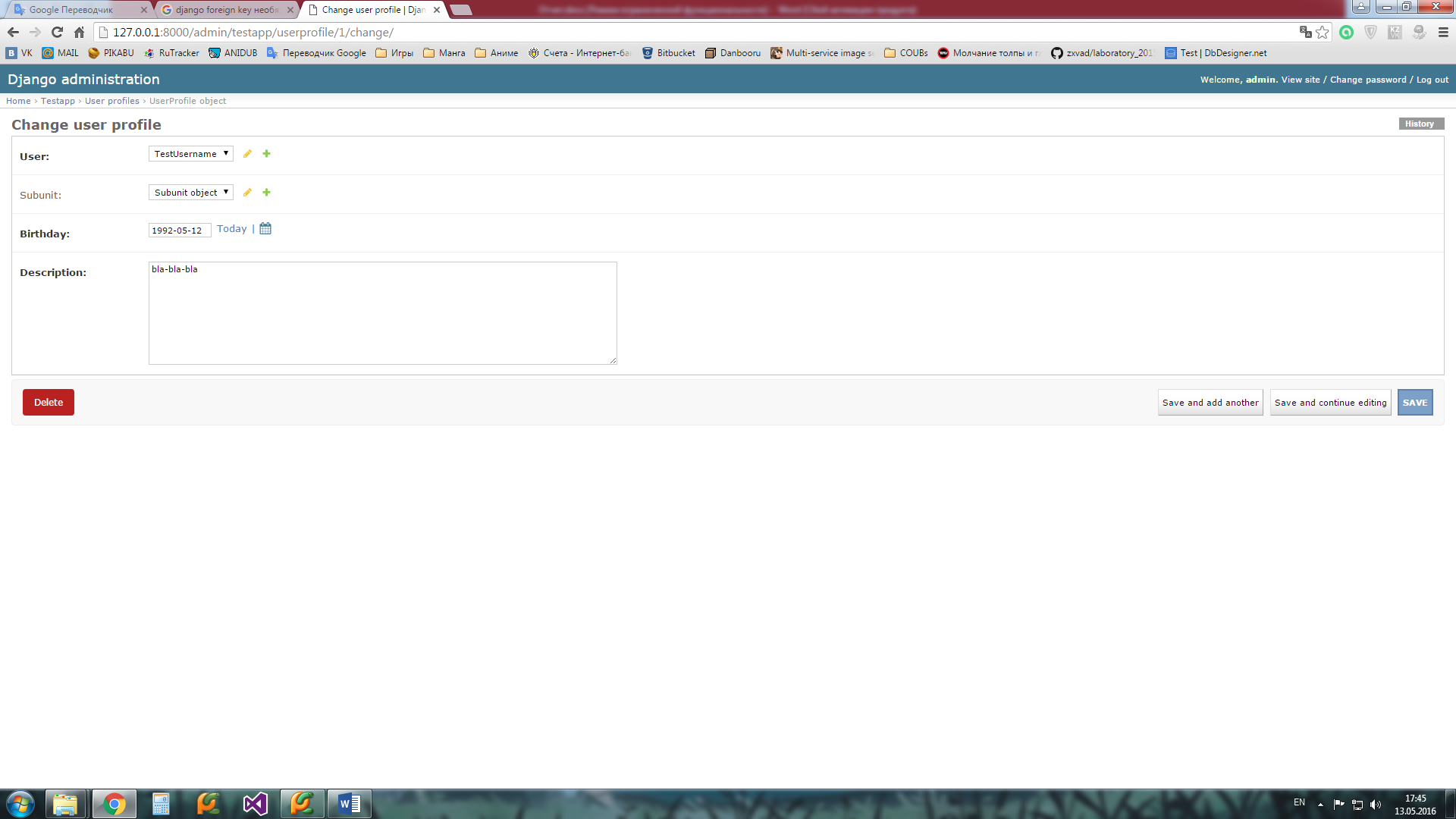


Рис. 4